US009280661B2

(12) **United States Patent**
Adjaoute

(10) **Patent No.: US 9,280,661 B2**
(45) **Date of Patent: Mar. 8, 2016**

(54) SYSTEM ADMINISTRATOR BEHAVIOR ANALYSIS

(71) Applicant: **Brighterion, Inc.**, San Francisco, CA (US)

(72) Inventor: **Akli Adjaoute**, Mill Valley, CA (US)

(73) Assignee: **Brighterion, Inc.**, San Francisco, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/634,786**

(22) Filed: **Feb. 28, 2015**

(65) **Prior Publication Data**

US 2015/0195300 A1 Jul. 9, 2015

Related U.S. Application Data

(63) Continuation-in-part of application No. 14/454,749, filed on Aug. 8, 2014, and a continuation-in-part of application No. 14/521,667, filed on Oct. 23, 2014, and a continuation-in-part of application No. 14/613,383, filed on Feb. 4, 2015.

(51) **Int. Cl.**

G06F 21/55	(2013.01)
H04L 29/06	(2006.01)
G06N 5/04	(2006.01)
G06Q 30/00	(2012.01)
G06Q 40/08	(2012.01)
G06Q 50/22	(2012.01)

(52) **U.S. Cl.**

CPC ***G06F 21/55*** (2013.01); ***G06F 21/552*** (2013.01); ***G06F 21/554*** (2013.01); ***G06N 5/048*** (2013.01); ***G06Q 30/0185*** (2013.01); ***G06Q 40/08*** (2013.01); ***G06Q 50/22*** (2013.01); ***H04L 63/1433*** (2013.01)

(58) **Field of Classification Search**

CPC G06F 21/55; G06F 21/552; G06F 21/554
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,377,354	A	12/1994	Scannell	
5,819,226	A	10/1998	Gopinathan	
6,161,130	A	12/2000	Horvitz	
6,272,479	B1	8/2001	Farry	
6,330,546	B1	12/2001	Gopinathan	
6,424,997	B1	7/2002	Buskirk, Jr.	
6,889,207	B2	5/2005	Slemmer	
7,036,146	B1 *	4/2006	Goldsmith	.... G06F 21/40 707/E17.007
7,406,502	B1	7/2008	Oliver	
7,433,960	B1	10/2008	Dube	
7,464,264	B2	12/2008	Goodman	
7,483,947	B2	1/2009	Starbuck	
7,562,122	B2	7/2009	Oliver	
7,668,769	B2	2/2010	Baker	
7,813,937	B1	10/2010	Pathria	
7,853,469	B2	12/2010	Maitland	
8,027,439	B2	9/2011	Zoldi	

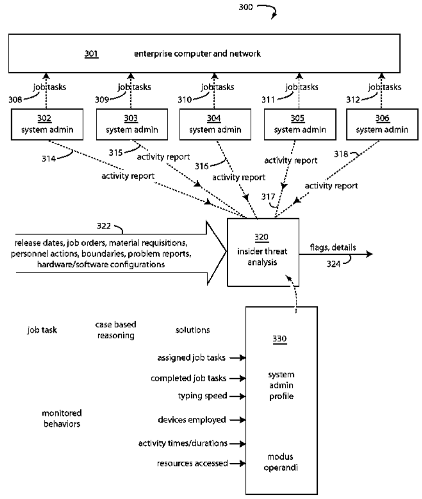
(Continued)

Primary Examiner — Kaveh Abrishamkar
Assistant Examiner — Ngoc D Nguyen
(74) *Attorney, Agent, or Firm* — Richard B. Main; Main Cafe

(57) ABSTRACT

A network computer system is protected from malicious attacks by its own system administrators by a large number of addressable and assignable smart-agents that are individually allocated to independently follow and represent those system administrators, the jobs those system administrated are assigned to work on, and the system resource tasks that such system administrators can employ in furtherance of the completion of a particular job.

1 Claim, 4 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

8,041,597	B2	10/2011	Li	
8,090,648	B2	1/2012	Zoldi	
8,458,069	B2	6/2013	Adjaoute	
8,484,301	B2	7/2013	Wilson	
8,548,137	B2	10/2013	Zoldi	
8,555,077	B2	10/2013	Davis	
8,572,736	B2	10/2013	Lin	
8,744,979	B2	6/2014	Sundelin	
2003/0009495	A1	1/2003	Adjaoute	
2003/0084449	A1 *	5/2003	Chane	H04N 5/44543 725/46
2004/0073634	A1 *	4/2004	Haghpassand	G06F 21/50 709/220
2007/0067853	A1	3/2007	Ramsey	
2007/0124246	A1	5/2007	Lawyer	
2010/0115610	A1	5/2010	Tredoux	
2011/0055196	A1	3/2011	Sundelin	
2011/0055264	A1	3/2011	Sundelin	
2013/0204755	A1	8/2013	Zoldi	
2014/0325643	A1 *	10/2014	Bart	H04L 63/1425 726/22

* cited by examiner

Fig. 1

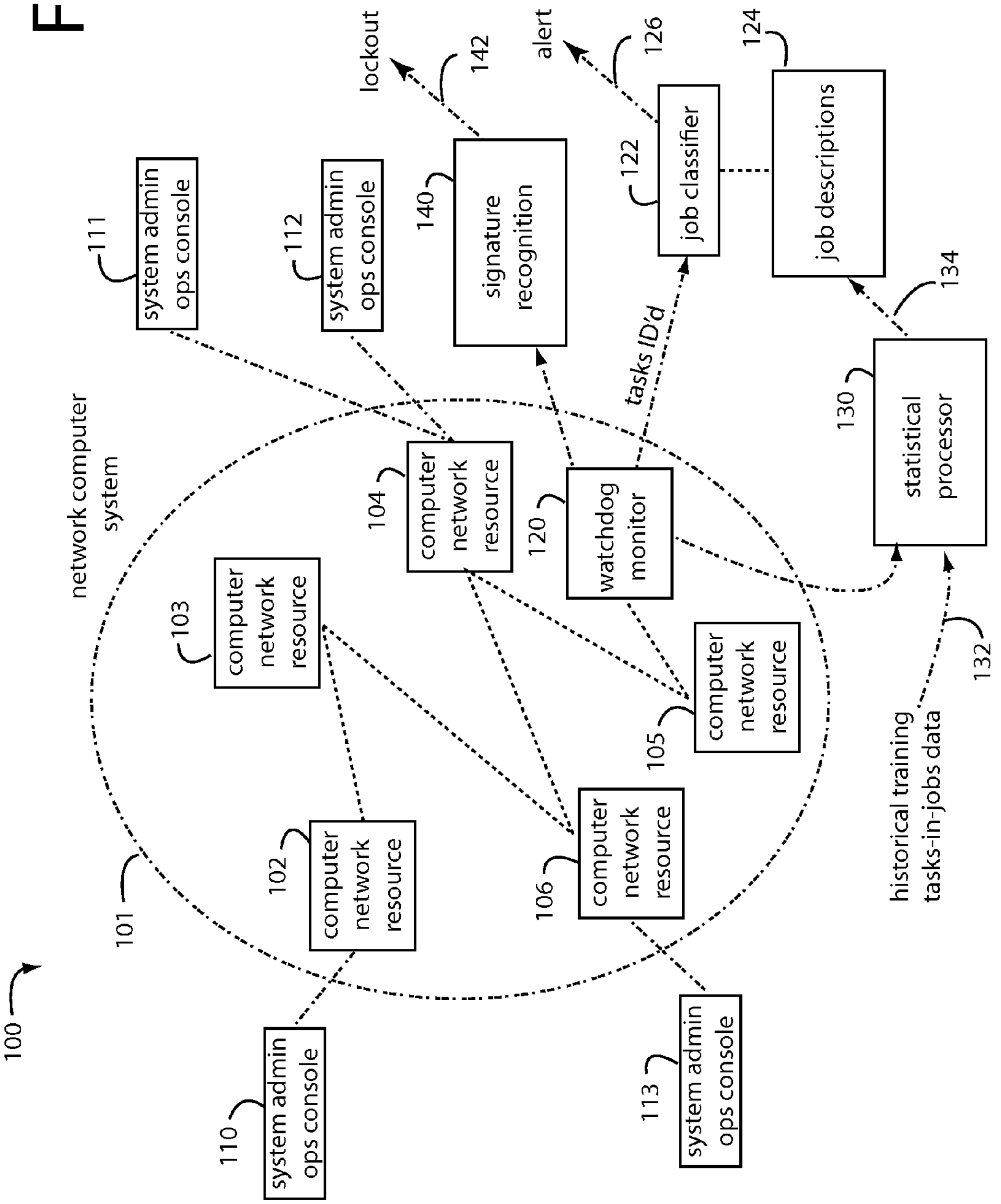

100
101
network computer system
102 computer network resource
103 computer network resource
104 computer network resource
105 computer network resource
106 computer network resource
110 system admin ops console
111 system admin ops console
112 system admin ops console
113 system admin ops console
120 watchdog monitor
122 job classifier
124 job descriptions
126
alert
130 statistical processor
132
historical training tasks-in-jobs data
134
140 signature recognition
142
lockout
tasks ID'd

Fig. 2

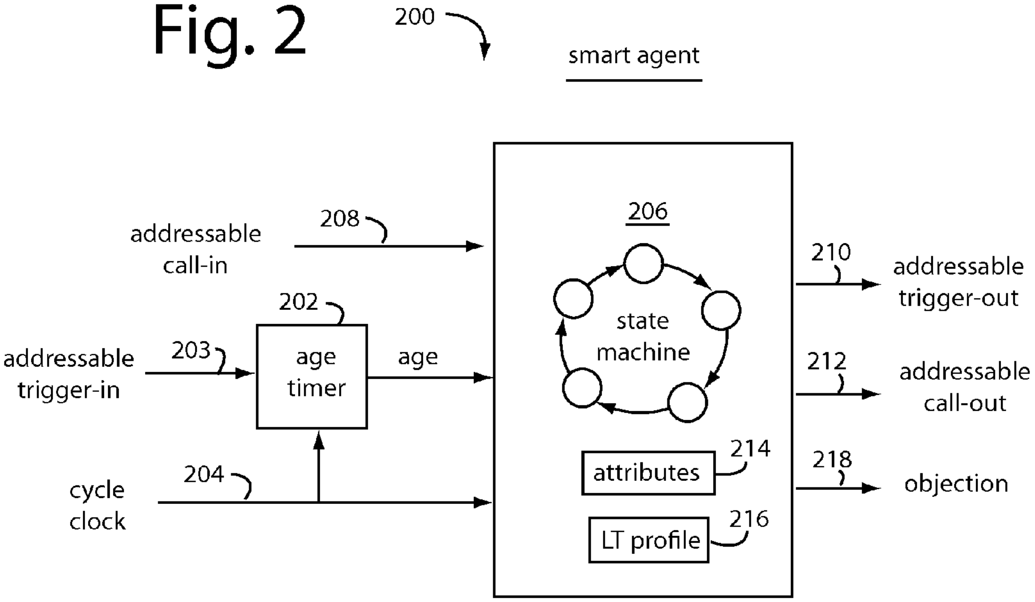

200
smart agent
206
state
machine
addressable
call-in
208
202
age
timer
addressable
trigger-in
203
age
cycle
clock
204
210
addressable
trigger-out
212
addressable
call-out
214
attributes
218
objection
216
LT profile

Fig. 3

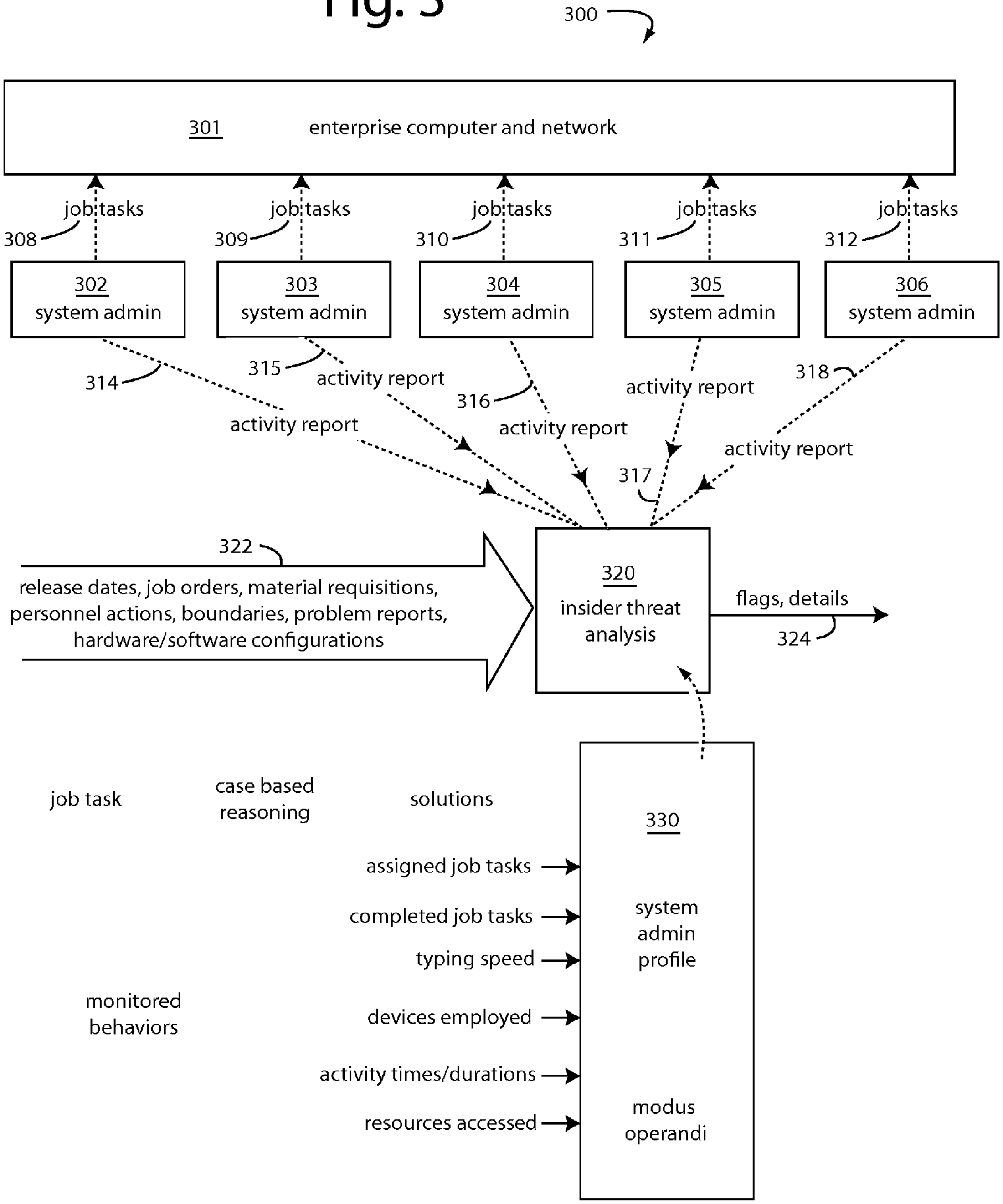

300
301 enterprise computer and network
job tasks
308
job tasks
309
job tasks
310
job tasks
311
job tasks
312
302 system admin
303 system admin
304 system admin
305 system admin
306 system admin
314
activity report
315
activity report
316
activity report
activity report
317
318
activity report
322
release dates, job orders, material requisitions, personnel actions, boundaries, problem reports, hardware/software configurations
320 insider threat analysis
flags, details
324
job task
case based reasoning
solutions
monitored behaviors
assigned job tasks
completed job tasks
typing speed
devices employed
activity times/durations
resources accessed
330
system admin profile
modus operandi

Fig. 4

activity report #	selected parameter							
	range-1	range-2	range-3	range-4	range-5	range-6	range-7	range-8
001				●				
002			●					
003							●	
004	●				●			
005			●					
006				●				
007				●				
008					●			
009			●					
010				●				
011					●			
012				●				
013			●					
014			●					

SYSTEM ADMINISTRATOR BEHAVIOR ANALYSIS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to artificial intelligence, and more particularly to business insider threats detectable by automated system administrator behavior analysis.

2. Description of Related Art

Big business is now routinely controlled by large computer systems and networks that operate on a scale so vast and quick as to be incomprehensible to average people. These computer systems and networks are, in turn, steered, monitored, and cared for by system administrators that have super access to all parts.

If the access privileges that system "admins" hold are abused, major economic and security damage can be caused all too silently to a company.

It has not been lost on investigators and analysts in general that evildoers and other perpetrators will behave in unusual ways, especially during the moments leading up to the commission of a crime. With computer systems, bad people often get away with impersonating real authorized users, but their odd behaviors will give them away.

So too with business insiders who have authorized access, but then abuse their privileges. But a special case is presented with system administrators because their behaviors are normally very chaotic, and patterns of normal behavior are absent even when their privileges are not being abused.

Twenty-three years ago Krishna Gopinathan, et al., proposed an automated system for fraud detection using predictive modeling. See, U.S. Pat. No. 5,819,226, filed Sep. 8, 1992. Neural networks were trained with historical and past transactional data, and used thereafter during operation to identify suspicious transactions based on learned relationships among the known variables. Their system periodically monitored a compliance metric of its fraud detection rate and its false positive rate. When their compliance metric fell below a minimum value, the system would automatically redevelop and adapt the fraud model.

What's not clearly disclosed is that only one model is ever developed and redeveloped from all the past transactional data to fit all the cardholders. Models are not individually created and assigned to track individual cardholders. That would work alright if the cardholders were fungible, but they're not, and each individual expresses sometimes unpredictable independence.

A "profile record" is created for cardholders by Krishna Gopinathan, et al., using the previous month's authorizations and cardholder data. Updates of individual cardholder activity use previous profile-record values and the previous month's authorizations and cardholder data. A "cascaded operation" adds a second neural network model trained only with transactions that achieved fraud scores from the first neural network model. Evidently cascades of three or four levels are possible.

Krishna Gopinathan, et al., provide a flowchart (FIG. 16) of a real-time system using the profile database. Upon receiving a merchant's request for authorization on a transaction 1602, the system obtains data for the current transaction 1603, as well as profile data summarizing transactional patterns for the customer 1604. It then applies this data to the stored neural network model 1605. A fraud score (representing the likelihood of fraud for the transaction) is obtained 1606 and compared to a threshold value 1607. Steps 1601 through 1607 occur before a transaction is authorized, so that the fraud score can be sent to an authorization system **1608** and the transaction blocked by the authorization system if the threshold has been exceeded. If the threshold is not exceeded, the low fraud score is sent to the authorization system **1609**. The system then updates a customer profile database 806 with the new transaction data **1610**. Thus, in this system, profile database 806 is always up to date (unlike the batch and semi-real-time systems, in which profile database 806 is updated only periodically).

The customer data from database 806 typically includes general information on the customer; data on all approved or declined transactions in the previous seven days; and, a profile record of data describing the customer's transactional pattern over the last six months. The general information on the customer typically includes customer zipcode; account open date; and card expiration date. Each profile record a profile database summarizes the customer transactional patterns as moving averages. The profile records are updated periodically, e.g., monthly, with all the customer transactions from the period.

Periodic redevelopment of the models makes it sound like the system can self-adapt. But their system constantly needs ever-improving training data that may not exist. The only diversity amongst the cardholders is in their respective transactions, not the fraud models being applied to them. Compliance only initially reaches optimum, and falls off immediately. Worse, each subsequent model redevelopment costs time offline. New kinds of fraud that evolve will disrupt such models because they're not equipped to evolve in tandem.

The short comings with these neural network models is needing to know what output is desired for each input before any training begins. Such can be very limiting. During training, if any of the desired outputs are left unknown for some input patterns, new incidences of fraud and abuse will go undetected in real-time. Detection that lags infection will exact a cost.

Neural networks, statistical modeling and profiling have been applied to fraud and abuse detection. But for them to be effective, they need a large database of cases in which fraud and abuse were detected. However, for this to work later the fraudulent methods and abuse must not have changed much. Such tools are impotent when the fraud and abuse either too closely resembles normal activity, or if it constantly shifts as the fraudsters adapt to changing surveillance strategies and technologies.

Conventional analytic solutions, even those that transaction to be non-hypothesis based, still operate within very rigid boundaries. They are either designed or tuned to look at various scenarios in such a way that they will only catch a limited range of the leakage problem. When something truly surprising happens, or a variation occurs that was not anticipated, systems based on such models fail to complete.

Modern systems need to be sophisticated, unsupervised, and learn as they go. New behaviors of fraud and abuse arise daily.

Conventional solutions to fraud have obtained only mediocre results. They lack scalability and always require high manual effort. We can do better.

SUMMARY OF THE INVENTION

Briefly, an artificial intelligence behavior analysis of system administrators in one embodiment of the present invention comprises software limited to the behavior analysis of system administrators within the confines of their particular job tasks, and operated on computer networks specifically improved and modified to have access to the resources involved in the job tasks assigned to the relevant system admins, and having storage to maintain profiles of their individual and respective behaviors in view of those job tasks.

Other and still further objects, features, and advantages of the present invention will become apparent upon consideration of the following detailed description of specific embodiments thereof, especially when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a functional block diagram of an improved network computer system with a group of several network-connected and interoperating computer network resources accessible by system administrators through system administrator consoles;

FIG. **2** is a functional block diagram of smart-agent embodiment of the present invention useful in the system of FIG. **1**;

FIG. **3** is a functional block diagram of a protected enterprise that includes an enterprise computer and network maintained and controlled by several authorized system admins; and

FIG. **4** is a table representing a series of activity reports #001-#014 filtered for a selected parameter.

DETAILED DESCRIPTION OF THE INVENTION

FIG. **1** represents an improved network computer system **100** with a group **101** of several network-connected and interoperating computer network resources **102-106** accessible by system administrators through system administrator consoles **110-113**. The network computer system **100** is vulnerable to malicious attacks by company insiders and fraudsters posing as system administrators (admins), e.g., via connections with the Internet. A principal objective of the present invention is to prevent or at least limit the damage that can be caused by purported system administrators.

The system administrator operator consoles **110-113** have privileged access to system resources **102-106** by way of selectable operating system tasks. These selectable operating system tasks include the typical system calls that conventional operating systems provide, e.g., file read/write, execute program, print, display, communication, etc.

The network computer system **100** is modified and adapted beyond a conventional arrangement of hardware to further include a watchdog monitor **120** connected to detect, record, and analyze which selectable operating system tasks each system administrator using a system administrator operator console **110-113** completes and the sequence of those task completes. A job classification processor **122** is connected to monitor and determine which, if any, of a plurality of system administrator jobs **124** by job description such individual system administrator's console completes and sequence of those particular task completes conforms to. For example, Job-A includes only tasks t1-t7, Job-B includes only tasks t8-t11, Job-C includes only tasks t12-t20, but Job-D also shares tasks t1-t7 but further also uses tasks t21-t33. A completing of any of tasks t34-t100 would be reason for concern because no authorized or described job includes any of those. Also, a sequence of tasks that danced over more than one job before finishing the last job is also troublesome.

JOB DESCRIPTION BY INCLUDED TASKS

Job	t1-t7	t8-t11	t12-t20	t21-t33	t34-t100
A	x				
B		x			
C			x		
D	x			x	

An security alert output **126** is constructed to enable management's attention to be called to suspicious system administrator activity if any individual system administrator's console completes and sequence of task completes do not conform to any one of the plurality of system administrator jobs **124**.

A statistical processor **130** receives task records from the watchdog monitor **120** during live operation, or historical training of tasks-in-jobs data **132** during training. The task records are related to which operating system tasks have been selected by system administrators to advance, or complete, particular system administrator jobs by a group of peers, and the statistical processor is programmed to set boundaries-of-inclusion **134** for which operating system tasks are used by a majority in the peer group to complete each particular system administrator job.

Thus a rogue system administrator can be distinguished from their peer group for highlighting and more intense scrutiny.

Some system administrator attacks follow recognizable patterns and have signatures that can be used to stop short any new such attack. A signature recognition processor **140** is also connected to receive task records from the watchdog monitor **120** and organizes them into single-source sequences according to the purported system administrator completing them. Signature recognition processor **140** compares these single-source sequences to particular sequences of operating system tasks that were infamously used by fraudsters and purported system administrators to compromise similar computer network systems. Signature recognition processor **140** can issue a lockout command **142** or flag the security alert output **126** if a match develops. A lockout output **142** triggers an automated logging-out of the offending purported system administrator and a quashing of their secure access credentials.

It would not be unusual for a conventional network computer system **101** to be trusted to keep secure the account records and personal details of ten million credit card accountholders. A typical system administrator has the system access privileges needed access all of these account records and personal details, and to engage in a massive data breach by spiriting off the sensitive data to evil doers. But, in improved network computer system **100** the operation system tasks and sequences that such would employ to carry off this crime would either have a recognizable signature to it or would not fit in any defined job.

In general, embodiments of the present invention include an artificial intelligence behavior analysis of system administrators. Most implementations require specialized software limited to the behavior analysis of system administrators within the confines of their particular job tasks. Such software is operated on computer networks specifically improved and modified to have access to the resources involved in the job tasks assigned to the relevant system admins, and has storage to maintain profiles of their individual and respective behaviors in view of those job tasks. It would also be sensible to put access to all of this outside the abilities of the system administrators being supervised, e.g., physical compartmentalization of the system's resources.

Recently, one employer looking to hire a System Administrator described the position as follows:

Essential Functions:

The System Administrator (SA) is responsible for effective provisioning, installation/configuration, operation, and maintenance of systems hardware and software and related infrastructure. This individual participates in technical research and development to enable continuing innovation within the infrastructure. This individual ensures that system hardware, operating systems, software systems, and related procedures adhere to organizational values, enabling staff, volunteers, and Partners.

This individual will assist project teams with technical issues in the Initiation and Planning phases of our standard Project Management Methodology. These activities include the definition of needs, benefits, and technical strategy; research & development within the project life-cycle; technical analysis and design; and support of operations staff in executing, testing and rolling-out the solutions. Participation on projects is focused on smoothing the transition of projects from development staff to production staff by completing operations activities within the project life-cycle.

This individual is accountable for Linux and Windows systems that support GIS infrastructure; Linus. Windows and Application systems that support Asset Management. Responsibilities include SA engineering and provisioning, operations and support, maintenance and research and development to ensure continual innovation.

SA Engineering and Provisioning

1. Engineering of SA-related solutions for various project and operational needs.

2. Install new/rebuild existing servers and configure hardware, peripherals, services, settings, directories, storage, etc. in accordance with standards and project/operational requirements.

3. Install and configure systems such as supports GIS infrastructure applications or Asset Management applications.

4. Develop and maintain installation and configuration procedures.

5. Contribute to and maintain system standards,

6. Research and recommend innovative, and where possible automated approaches for system administration tasks. Identify approaches that leverage our resources and provide economies of scale.

Operations and Support

7. Complete daily system monitoring, verifying the integrity and availability of all hardware, server resources, systems and key processes, reviewing system and application logs, and verifying completion of scheduled jobs such as backups.

8. Complete regular security monitoring to identify any possible intrusions.

9. Complete daily backup operations, ensuring all required file systems and system data are successfully backed up to the appropriate media, recovery tapes or disks are created, and media is recycled and sent off site as necessary.

10. Complete regular file archival and purge as necessary.

11. Create, change, and delete user accounts per request.

12. Provide Tier III/other support per request from various constituencies. Investigate and troubleshoot issues.

13. Repair and recover from hardware and software failures. Coordinate and communicate with impacted communities.

Maintenance

14. Apply operating system patches and upgrades on a regular basis, and upgrade administrative tools and utilities. Configure/add new services as necessary.

15. Upgrade and configure system software that supports GIS infrastructure applications or Asset Management applications per project or operational needs.

16. Maintain operational, configuration, or other procedures.

17. Complete periodic compliance reporting to support capacity planning.

18. Complete ongoing compliance tuning, hardware upgrades, and resource optimization as required. Configure CPU, memory, and disk partitions as required.

19. Maintain data center environmental and monitoring equipment.

Knowledge/Skills:

1. Bachelor (4-year) degree, with a technical major, such as engineering or computer science.

2. Systems Administration/System Engineer certification in Unix and Microsoft.

3. Four to six years system administration experience.

Complexity/Problem Solving:

1. Position deals with a variety of problems and sometime has to decide which answer is best. The question/issues are typically clear and requires determination of which answer (from a few choices) is the best.

Discretion/Latitude/Decision-Making:

1. Decisions normally have a noticeable effect department-wide and company-wide, and judgment errors can typically require one (o two weeks to correct or reverse.

Responsibility/Oversight—Financial & Supervisory:

1. Functions as a lead worker doing the work similar to those in the work unit; responsibility for training, instruction, setting the work pace, and possibly evaluating compliance.

2. No budget responsibility

Communications/Interpersonal Contacts:

1. Interpret and/or discuss information with others, which involves terminology or concepts not familiar to many people; regularly provide advice and recommend actions involving rather complex issues. May resolve problems within established practices.

2. Provides occasional guidance, some of which is technical.

Working Conditions/Physical Effort:

1. Responsibilities sometimes require working evenings and weekends, sometimes with little advanced notice.

2. No regular travel required.

System administrators can therefore come and go, night and day, access resources they never accessed before, and can pretty much have a free run of an entire business enterprise without raising suspicions. Eric Snowden recently took advantage of his system administrator capabilities to cause the United States Government a major embarrassment through his now-famous data breach.

Many other jobs related to system administration may be separate positions within a computer support or Information Services (IS) department. Database administrators (DBA) maintain a database system, and is responsible for the integrity of the data and the efficiency and compliance of the system. Network administrators maintain network infrastructures such as switches and routers, and diagnose problems with these or with the behavior of network-attached computers. Security administrators are specialists in computer and network security, including the administration of firewalls and other security devices, and consult on general security measures. Web administrators maintain webserver services (such as Apache or IIS) that allow for internal or external access to web sites. Tasks include managing multiple sites, administering security, and configuring necessary components and software. Responsibilities may also include software change management. Computer operators do routine maintenance and upkeep, such as changing backup tapes or replacing failed drives in a redundant array of independent disks (RAID). Such tasks usually require physical presence in the room with the computer, and may require a similar level of trust as system admins, since they have access to sensitive data. Postmasters administer mail servers. Storage (SAN) Administrators create, provision, add or remove Storage to/from Computer systems. Storage can be attached locally to the system or from a storage area network (SAN) or network-attached storage (NAS). The administrator can also create file systems from newly added storage. Often, those who begin as a member of the technical support staff or a computer operator, will be promoted to a "sysadmin" position after gaining wider experience on the job.

All of these insider positions involve a great deal of trust and risk exposure by business enterprises and even governments and their militaries and other agencies and departments.

By one account, system administrators are responsible for the technical design, planning, implementation, and the highest level of compliance tuning and recovery procedures for mission critical enterprise systems. They serve as technical experts in the area of system administration for complex operating systems. They recommend the redesign and configuration of operating systems and system applications. System administrators investigate and analyze system requirements feasibility and develop system specifications. They identify methods and solutions, and provide project leadership and management. System administrators often provide comprehensive supervision of operations staff.

Typical Duties and Responsibilities of a System Administrator

Manages the day-to-day operations of the host computers by monitoring system compliance, configuration, maintenance and repair. Ensures that records of system downtime and equipment inventory are properly maintained. Applies revisions to host system firmware and software. Works with vendors to assist support activities.

Develops new system and application implementation plans, custom scripts and testing procedures to ensure operational reliability. Trains technical staff in how to use new software and hardware developed and/or acquired.

Supervises Operations staff including hiring, training, evaluating and disciplining. May guide or provide work direction to technical staff, contract staff and/or student employees. Determines appropriate coverage for all hours of operation.

Completes troubleshooting as required. As such, leads problem-solving efforts often involving outside vendors and other support personnel and/or organizations.

Establishes, maintains and manages users Unix accounts. Installs, modifies and maintains systems and utility software on server computer systems. Provides server support related to other software.

Establishes guidelines and methods for the installation and management of the host computer operating systems, disk arrays, fiber channel switches, tape libraries and other components.

Ensures high availability and acceptable levels of compliance of mission critical host computer resources.

Develops procedures to maintain security and protect systems from unauthorized use, acts of nature and user abuse.

Develops procedures, programs and documentation for backup and restoration of host operating systems and host-based applications.

Develops and coordinates project directions and schedules to maximize benefits and minimize impacts on the customer

organizations. Provides leadership in planning and implementation of projects for computer operations and enterprise systems administration.

Develops toots, procedures, and training sessions for Operations. Client Support and

Systems Development staff to assist with work.

Manages the data center and computer host systems including hardware, software and equipment such as air-conditioning system, UPS (uninterrupted power system) and fire protection system.

Stays current with technological developments in systems administration technology and recommends ways to take advantage of new technology.

On the computer network system itself, a system administrator's technical responsibilities might further include:

- Analyzing system logs and identifying potential issues with the computer systems;
- Introducing, integrating, and testing new technologies into existing data center environments;
- Routine audits of systems and software.
- Applying operating system updates, patches, and configuration changes;
- Installing and configuring new hardware and software.
- Adding, removing, or updating user account information, resetting passwords;
- Answering technical queries and assisting users;
- Security;
- Documenting system configurations;
- Troubleshooting reported problems;
- Tuning system compliance;
- Confirming that the network infrastructure is up and running;
- Configuring, adding, and deleting file systems.

A monitor and log that tracks an individual system administrator should see these kinds of things going on (and probably nothing else outside their formal authority). For example, large print jobs or transfers of files or databases to USB drives would be unusual and hard to justify.

In another example, the routine duties of a database administrator include:

- Installing and upgrading the database server and application tools;
- Allocating system storage and planning future storage requirements for the database system;
- Modifying the database structure, as necessary, from information given by application developers;
- Enrolling users and maintaining system security;
- Ensuring compliance with database vendor license agreement;
- Controlling and monitoring user access to the database;
- Monitoring and optimizing the compliance of the database;
- Planning for backup and recovery of database information
- Maintaining archived data;
- Backing up and restoring databases;
- Contacting database vendor for technical support;
- Generating various reports by querying from database as per need.

So a monitor and log that tracks an individual database administrator should see these kinds of things going on.

Individuals develop their own styles and habits. They do things in particular sequences, use a kit of favorite tools, write scripts they trust and have used before, and react in repeatable ways to various stimuli.

In an embodiment of the present invention, a particular system administrator's jobs are artificially categorized in the following way:

Job Code	Job Description	Frequency
A	Analyzing system logs and identifying potential issues with the computer systems	daily, in background
B	Introducing, integrating, and testing new technologies into existing data center environments	as needed
C	Routine audits of systems and software	periodic, maybe monthly
D	Applying operating system updates, patches, and configuration changes	as such are released
E	Installing and configuring new hardware and software.	as received, and traceable to purchase orders
F	Adding, removing, or updating user account information, resetting passwords	as requested, and in part related to human resources activities
G	Answering technical queries and assisting users	as requested, and linkable to particular users
H	Security	ongoing
J	Documenting system configurations	as configuration changes occur, and traceable to work orders
K	Troubleshooting reported problems	as requested, and linkable to particular users
L	Tuning system compliance	random
M	Confirming that the network infrastructure is up and running	periodic and should be harmless
N	Configuring, adding, transferring, copying, and deleting file systems	who, what, where, when, and why all need to be within bounds

Every system administrator's days, weeks, and months will be divided amongst these jobs, and some may have the personal freedom to do them in any order, sequence, or priority that they choose. Herein is a first aspect of their individual and peer group behaviors that can be tracked, profiled, and recognized.

Jobs are assigned to system administrators and often well-planned even months in advance. As a group, the system administrators will engage various tasks they have in a toolkit in order to complete each job. Not every system administrator will approach every job with the same task completes and they may vary in the sequences of task completes. But over time patterns will emerge of normal behavior and certainty training data can be used to initiate what is normal behavior.

What is meant by "task" herein is a particular operating system activity like reading a file, writing a file, copying a file, downloading a file, executing a script, printing a page, accessing a sensitive data file, running a program, displaying a graphic, etc.

During live watchdog operation, the technical behaviors of a company's system administrators are monitored by embodiments of the present invention. More precisely, the particular tasks they complete for any reason are tracked as are the sequences of their use.

The various jobs a system administrator is authorized to do have well-defined limits, and the tasks employable to complete each of those jobs are also limited by job descriptions. So the use of a task not in any job description, or the sequencing of tasks not staying within a single job description until the job is completed are suspect.

System administrators normally stay on a particular job until it's completed and before moving on to a next. So bouncing around by using tasks unique to more than one job is questionable.

So, if a particular system administrator is observed by a watchdog monitor to have completed a task, or a sequence of tasks, not found in any one authorized job description, then that particular system administrator warrants more intense scrutiny and may even require automated deactivation.

For example, the job description of a particular "JOB-A" expects system administrators to use tasks "T1-T7". In watchdog mode, eight system administrators 1-8 were observed to employ the following tasks.

JOB-A									
	T1	T2	T3	T4	T5	T6	T7	T8	T9
SA1	1	1	0	0	0	1	0	1	1
SA2	1	1	0	0	0	0	0	0	1
SA3	1	1	0	1	1	0	0	0	1
SA4	0	0	0	0	0	0	0	1	0
SA5	1	0	0	1	1	0	0	0	1
SA6	1	1	0	1	1	1	0	1	0
SA7	1	1	0	1	1	1	0	0	0
SA8	1	1	0	1	1	1	0	0	1

The technical behavior of System administrator 4 is odd and out-of-character compared to their peers, System administrator 1-3 and 5-8. Specifically, the peer group all employed task t1, most used task t2, a majority used task t3, t4, all used task t4 if they used t3, etc.

A case-based-reasoning (CBR) system can be employed to help analyze the behavior of system administrators. A general purpose fraud classifier, one-size-fits-all for system administrators is not going to be very useful. A better implementation is based on the so-called "smart-agents" of the present inventor, Dr. Akli Adjaoute, as are described in his earlier United States Patent Applications. These are all incorporated by reference herein in full, and in particular enumerated by those Applications herein claimed as parent to this Continuation-In-Part.

USPTO FILINGS OF AKLI ADJAOUTE INCLUDED-BY-REFERENCE		
USPTO appl. no.	Official Filing Date	Title
14/180,370	14 Feb. 2014	Multi-Dimensional Behavior Device ID
14/243,097	2 Apr. 2014	Smart Analytics For Audience-Appropriate Commercial Messaging
14/454,749	8 Aug. 2014	Healthcare Fraud Preemption
14/514,381	15 Oct. 2014	Artificial Intelligence Fraud Management Solution
14/517,863	19 Oct. 2014	User Device Profiling In Transaction Authentications
14/525,273	28 Oct. 2014	Data Breach Detection
14/521,667	23 Oct. 2014	Behavior Tracking Smart-agents For Artificial Intelligence Fraud Protection And Management
14/521,386	22 Oct. 2014	Reducing False Positives With Transaction Behavior Forecasting
14/520,361	22 Oct. 2014	Fast Access Vectors In Real-Time Behavioral Profiling
14/517,771	17 Oct. 2014	Real-Time Cross-Channel Fraud Protection
14/522,463	23 Oct. 2014	Smart Retail Analytics And Commercial Messaging
14/517,872	19 Oct. 2014	Healthcare Fraud Protection And Management
14/613,383	4 Feb. 2015	Artificial Intelligence For Context Classifier

In various embodiments of the present invention, a smart-agent with CBR is virtually "attached" and assigned to every individual system admin, job, and task. Storage room for their respective profiles are maintained in secure memory.

The term "smart-agent" has had some use in the prior art, but what is meant here by "smart-agent" is altogether different. Prior Patent Applications by the present inventor, Dr. Akli Adjaoute, have described what is meant here in a number of different ways.

Referring now to FIG. **2**, each smart-agent **200** is addressable and has a timer **202** can be triggered into life with an addressable trigger-in **203** and begin aging tick-by-tick with a cycle clock **204**. A state machine **206** can be addressably called into action like a "call" to a subroutine with an addressable call-in **208**. An addressable trigger-out **210** can trigger into life other smart-agents. An addressable call-out **212** can call into action other smart-agents as if they were addressable subroutines. A list of attributes **214** describes, in an exemplary instance here, the particular tasks employed by this particular job, or the tasks that a particular system administrator can employ. A long term (LT) profile **216** is a memory log of the past activities that this smart-agent was involved in, and is able to develop a behavior profile of what is "normal" behavior for this entity.

An objection **218** can issue by the state machine **206** if the instant behavior for this entity seems abnormal, or if an age timeout **220** occurs before the state machine has run or finished in response to an addressable call-in **208**.

Activity reports **220** are cleaned up, filtered for the particular smart-agent **200**, and used to build LT profile **216**. As each report comes in its information is inspected by state machine **206** to see if the activity was expected, normal, timely, respected priorities, etc. For example, if the activity was the running of a task.

Once an addressable call-in **208** is received, the state machine **206** will typically consult the attributes **214** to see what other addressable triggers-out **210** and addressable calls-out **212** should issue and in which clock cycles. For example, if a Job-A requires tasks t1-t7 to be run, then the Job-A smart-agent will trigger all seven of the T1-T7 smart-agents. If they timeout (age is too old) without having been employed in a call by the system admin, then the ones who weren't called into action will issue objections.

Here, in this Application, an individual smart-agent **200** is spawned and attached to every identifiable system admin, job, and task. Each such smart-agent has its own characteristic attributes, e.g., a job smart-agent will have task attributes corresponding to every task that this particular job has called, should call, or should not call. The tasks it calls can have a priority order, and that would be another attribute and another smart-agent. The various smart-agents are interconnected, interrelated and each can be randomly accessed and consulted.

For example, any job smart-agent can have its LT profile **216** accessed to see who has called it, triggered it, it has called, it has triggered, etc. It can further be queried as to its attributes **214**. It is therefore as easy to query what jobs have been done by which system administrators as it is to query which system administrators have done which jobs.

A CBR case consists of a problem, a previous solution that worked, and remarks about how the solution was derived. Case-based reasoning can be formalized as a four-step process:

Retrieve	For each target problem, cases are retrieved from memory relevant to solving it.
Reuse	The solution is mapped from the previous case to the target problem and may involve adapting the solution to fit the new situation.
Revise	The new solution is tested and, if necessary, revised.
Retain	After a solution has been used successfully on the target problem, the resulting experience is stored as a new case in memory.

Herein, a case comprises a system administrator's job task and the solutions comprise what particular system administrators did to do that job task. (There being many ways to solve a problem or do a job that will express the personalities involved.)

Each system administrator activity report comes in like the payment transactions that are described in the Applicant's previous patent applications involving payment fraud detection. Here, system administrator activities can be fraudulent, suspicious, or apparently acceptable.

Referring now to FIG. **3**, a protected enterprise **300** includes an enterprise computer and network **301** maintained and controlled by, e.g., five authorized system administrators **302-306**. Each system administrator **302-306** is assigned jobs that they then complete as job-related tasks **308-312** directed within the protected enterprise computer network **300**. Activity reports **314-318** related to each system administrator **302-306** automatically issue to an insider threat analysis device **320**.

The insider threat analysis device **320** is an embodiment of the present invention that can be integrated into an otherwise already existing and conventional operation. It will ordinary include hundreds if not thousands of uniquely assigned smart-agents **200**. In an optimum configuration, insider threat analysis device **320** receives automated software update release dates, company official job orders, approved material requisitions, confidential personnel actions, security boundaries, customer problem reports, protected enterprise computer network hardware/software configurations, and other corroborating input information **322**.

A management information system (MIS) is a computerized database of financial information organized and programmed in such a way that it produces regular reports on operations for every level of management in a company. It is also usually possible to obtain special reports from MIS systems. Many of the job tasks **308-312** operating on protected enterprise computer network **300** will have corresponding MIS facts that can be used to expect, verify, corroborate, confirm, or otherwise validate an activity report **314-318**. Activities that should have been expected, verifiable, supportable, confirmable, attestable, or otherwise provable, but were not, are reasons to output flags and details **324**.

In a unique and novel aspect, activity reports **314-318** that reflect behaviors outside "normal" behavior for the corresponding system administrator **302-306** will be more telling reasons to output flags and details **324**. The insider threat analysis device **320** could be expected to provide meaningful output flags and details **324** even if no information at input **322** was forthcoming.

The information must be transformed into a format that insider threat analysis device **320** can operate on. Specifically, data is categorized and tallies for each category are stored in memory for comparisons later. For example, software release dates can be categorized by the number of days the software has been released, or by the month/year of its release. Job task categories were categorized above, and activities in activity reports can be categorized as well.

In general, an insider threat analysis device embodiment of the present invention is connected into an enterprise's protected computer network such that a series of activity reports of system administrators can be securely collected and securely stored in memory. Particular aspects of the series of activity reports are selected, categorized, and the results tabulated. Each corresponding individual system administrator is profiled by a smart-agent with memory storage by the tabulated results obtained over time from many activity reports related to an individual system administrator. In FIG. **3**, this is represented by system administrator profile **330**.

A type of biometric is important to be sensed for each corresponding individual system administrators from indications discernable from any of the activity reports. Various devices for sensing a biometric are able to detect at least one of a keyboard typing speed, system tools employed, scripts used, time of activity and duration, and the particular workstations used.

At least one biometric previously sensed is compared to a corresponding one currently being sensed. A behavior processor then determines if the biometrics sensed were unlikely to have been produced by the same original system administrator. If some aspect of a system administrator's behavior is unusual given the profile maintained for them by the smart-agent, a misbehaving-system-admin security alert is output. Such can occur if an activity report concludes in tabulated results for a corresponding individual system administrator that deviate substantially from said profile of tabulated results obtained over time.

Considering now FIG. **4**, a series of activity reports #001-#014 are filtered for a selected parameter, e.g., typing speed at the system administrator's keyboard. Such would be tracked as an attribute of the smart-agent for that system administrator and maintained in a long term profile.

In order to save having to compute each parameter for comparison in realtime, each selected parameter is categorized into ranges and the ranges are populated into a tabulation maintained in memory. The whole represents a simple one-dimension profile useful to a smart-agent. It can be seen by simple visual inspection that the vast majority of selected parameter measurements categorize to range-3 and range-4. Activity report #003 produced a range-7 tabulation that is the most extreme. An adjustable trigger can be set so the deviation is enough to produce a misbehaving-system-admin security alert output.

In general, embodiments of the present invention have three analytical objectives:

1. Analyze the general behavior of the system administrator;
2. Analyze the activities of the system administrator and compare their pertinence to each job queued (lists of Jobs) to be completed by the system administrator; and
3. Intrusion detection.

Intrusion detection systems (IDSs) can be divided into two categories, network-based and host-based. Network-based systems (NIDS) listen on the network and monitor individual data packets flowing through a network. NIDS's often require dedicated hosts and special equipment vulnerable to network attacks. Host-based intrusion detection systems (HIDS) deal with each individual host and monitor the system in real-time to detect abnormal behavior and other activities such as repeated failed access attempts, changes to critical system files log files, etc. HIDS clients can be installed on every host on a network and tailored to the specific host's configuration. They then watch for anomalies and abuses of the systems.

Embodiments of the present invention combine smart-agents, case based reasoning, business rules, real-time and long-term profiling, and fuzzy logic.

Each system administrator job is assigned to and represented by a smart-agent, as are every task and task that are expected to be completed are associated as attributes. Each activity/task represented by a smart-agent is automatically linked to all the job smart-agents that they need to do.

Each job is linked to all the potential activities that are related to them and each activity/task is interconnected with all the system administrator jobs that also use those tasks and activities.

A job clock is started at the beginning of each job. All the smart-agents corresponding to that job are triggered or "born", and are assigned an age attribute of "1" and a completion time attribute. Every time an activity or task is completed, all the jobs that normally can benefit from such tasks are also triggered. The job clock thereafter advances the smart-agents cycle-by-cycle. The age of any of the jobs so triggered can be read and will be the number of job clock cycles since its activation by a system administrator. If there is a job problem, an objection will issue.

There are three basic types of objections:

(1) Task-omission objections are triggered by tasks that should have been completed by the system administrator but were not;

(2) Uncalled-for task objections are triggered by any task that should not have been used, because it is not related to the job or the current situation, but was completed anyway by the system administrator. Excess objections are also triggered by prematurely completed tasks that required a prerequisite, but missing, activity;

(3) Priority task-order objections are triggered by tasks that should have been completed in order sooner, or critical tasks related to the job were not completed, or less critical tasks are being completed first, or task completions are not following the proper order.

The number of, and the importance of, and the criticality of the other tasks that should have been completed before the offending task that was chosen by the system administrator are combined to give a weight that is then used to compute a compliance level.

As soon as any job is assigned to a particular system admin, all the corresponding tasks that must be completed for that job are triggered and the job clock runs. The tasks that follow must be completed in proper time and proceed in proper order.

When a system administrator is completing tasks not related to the job or not in the right order a security alert will be triggered. Any critical task not related to the job will generate a security alert if the system administrator is trying to complete the action/task.

If the system administrator is following a path, or sequence of actions that could be related to the current job, the system may accept this behavior as normal. But, no critical tasks related to the current job can be missed, and no critical actions should be undertaken that are not related to the current job.

If the system administrator is supposed to complete a specific job then all the task smart-agents related to this job will be expecting to be triggered according to a fixed order (age).

When a job smart-agent is triggered to start, all the task smart-agents anticipated to begin straightaway are in turn triggered and initialized with an age of "1". If the system administrator in fact completes all the expected tasks by age-1, then the work that was done is considered to be in compliance. Otherwise, all the task smart-agents that were supposed to run in the first cycle but were stood up will file a objection. A security alert is sent to a judging module that scores the system administrator's work compliance. The offending system administrators can either be allowed to continue, or they can be disconnected and sidelined. The severity of their offenses makes the difference.

Not all system administrators will complete all the expected tasks related to a particular job in the same order. If a few of the tasks that were expected to transpire in any cycle did not, the system administrator may nevertheless have some latitude in getting to them later.

The judging module will return a confidence score. If the score exceeds a minimum confidence threshold, the system administrator will be allowed to continue to the next cycle and all the activities expected that were not completed will increase their age. When the system administrator completes the next new task, this task too will assess its relation to the job at hand.

The kit of tasks that can be combined to do a job is automatically learned from what is the norm for the group.

Some jobs may be completed in more than one way. Independent system administrators will often use their own mix of favorite techniques. For example, given seven different tasks (t1-t7) and eight individual system administrators (SA1-SA8), the table below represents the task they chose to complete a typical job.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	total
SA1	1	1	0	0	0	1	0	1	1	5/9
SA2	1	1	0	0	0	0	0	0	1	3/9
SA3	1	1	0	1	1	0	0	0	1	5/9
SA4	0	0	0	0	0	0	0	1	0	1/9
SA5	1	0	0	1	1	0	0	0	1	4/9
SA6	1	1	0	1	1	1	0	1	0	6/9
SA7	1	1	0	1	1	1	0	0	0	5/9
SA8	1	1	0	1	1	1	0	0	1	6/9

Clearly, something is definitely odd about SA4, and SA2 is suspect. From this table the tasks can be inversely related to the corresponding system administrators that used them.

	SA1	SA2	SA3	SA4	SA5	SA6	SA7	SA8	total
T1	1	1	1	0	1	1	1	1	7/8
T2	1	1	1	0	0	1	1	1	6/8
T3	0	0	0	0	0	0	0	0	0/8
T4	0	0	1	0	1	1	1	1	5/8
T5	0	0	1	0	1	1	1	1	5/8
T6	1	0	0	0	0	1	1	1	4/8
T7	0	0	0	0	0	0	0	0	0/8
T8	1	0	0	1	0	1	0	0	3/8
T9	1	1	1	0	1	0	0	1	5/8

Here, t1 was employed by all except SA4; t2 was employed by all except SA4 and SA5. T3 and t7 were ignored by all system administrators.

Such allows the following rule to be defined:

A first list L1 of system administrators "I" is said to be more global than a second list L2 of system administrators "J" if all its system administrators also completed all the tasks in list "I", and is noted as [[ ]], in our example:

L1 [[ ]] L2

L1 Non [[ ]] L6.

The notion of being global is transitive

L1 [[ ]] L2 et L2 [[ ]] L8 - - - à L1 [[ ]] L8

The list of tasks is sorted according to the number of system administrators that use them. If two lists are identical (4 et 5), only one is kept, but remembering the redundancy is important.

The preferred tasks are on top.

Creation of the Clusters of Tasks

Creation of the Analysis Matrix CLU (I, J)

Line=tasks

Column=Tasks+SAD+Test

If TASK(i,j) [[ ]] TASK(i+1 j) then CLU(i i+1)=1 & CLU(i+1 test)=CLU(i+1 test) U CLU(i test).

Task8 include itself and in task 6 as a result

CLU(Task8 Task8)=1

CLU(Task8 Task6)=1

CLU(task6 test)=1,4,6

Each line that includes another line will excluded unless CLU(Task, ADM) is different to CLU (task, test)

	task-8	task-4	Task 5	task-6	task-9	task-2	task-1	SAD	C
task-8	1			1				1, 4, 6	
task-4		1	1				1	3, 5, 6, 7, 8	
Task 5								3, 5, 6, 7, 8	
task-6				1				1, 4, 6, 7, 8	1, 4, 6
task-9					1		1	1, 2, 3, 5, 8	
task-2						1	1	1 ,2, 3, 6, 7, 8	
task-1								1, 2, 3, 5, 6, 7, 8	3, 5, 6, 7, 8: 1, 2, . . .

The following four clusters are generated:

1. {task8, task6}
2. {task4, task5, task1}
3. {task9, task 1}
4. {task2, task1}

Each system administrator will perform one or more clusters. All task of a cluster must be performed.

Data Mining and neural networks use a large, frequently updated database of known attack signatures to construct a decision tree for detecting attacks with known signatures.

Real-time Profiling analyzes the current activities of the system administrator will be compared the long-term profiles of the system administrator learned from the previous activities by the . . . week, month, year.

An important component of case based reasoning is the case archives where the previously experienced tasks and jobs are stored with a list of actions, tasks, and activities in one case. Each line of a case describes one feature, action, and/or task in the case. When a job is listed in the tasks assigned to a system admin, the cases related to this job are activated. This monitors the activities of the system administrator and measures any similarity between the matching features of the cases related to the selected cases and the actions activities completed by the system administrator. The returned cases are ranked according to their degrees of similarity to the given problem.

Although particular embodiments of the present invention have been described and illustrated, such is not intended to limit the invention. Modifications and changes will no doubt become apparent to those skilled in the art, and it is intended that the invention only be limited by the scope of the appended claims.

The invention claimed is:

1. A method of protecting a network computer system from the malicious acts of its own system administrators, comprising:

providing privileged access to system resources by system administrators to a computer network including through system administrator operator consoles by way of selectable operating system tasks;

detecting, recording, and analyzing with a watchdog monitor which said selectable operating system tasks each system administrator employs at a system administrator operator console to any particular task, and their sequence;

determining with a job classification processor connected to monitor and determine which, if any, of a plurality of system administrator jobs an individual system administrator's console appears to be following by the tasks being completed and the sequence in which the tasks are being completed;

calling attention to system administrator activity with a security alert output if any individual system administrator's console is used to complete any individual task or any sequence of tasks that do not conform to any one of the plurality of system administrator jobs;

independently following representations of individual system administrators with a plurality of smart-agents that record the jobs those system administrators can be assigned to work on, and that list the system resource tasks that such system administrators are preauthorized to employ;

maintaining a computed confidence score associated with each system administrator that represents a probability the system resource tasks that corresponding system administrators employed were preauthorized and conform in their sequences to a particular job;

triggering a smart agent timer with an addressable trigger-in to begin aging tick-by-tick with a cycle clock;

calling a state machine into action with an addressable call-in;

triggering other smart agents with an addressable trigger-out;

calling into action other smart-agents with an addressable call-out;

listing any attributes that describe particular tasks employed by a particular job, or the tasks that a particular system administrator is preauthorized to employ;

logging into a long term (LT) profile memory the past activities that a smart-agent was involved in, and later are used to contribute to a normal-behavior profile for an entity;

issuing an a objection with the state machine if an instant behavior for the entity is abnormal, or if an age timeout occurs before the state machine has run or finished in response to an addressable call-in;

inputting activity reports filtered for particular smart-agents, and used to build the long term profile;

inspecting the activity reports with the state machine in a determination of whether the activity reported was expected, normal, timely, and respected priorities;

consulting the attributes in a determination of what other addressable triggers-out and addressable calls-out should issue and in which clock cycles, and

issuing objections from a task smart-agent if a timeout occurs without having it having been employed in a call by the system administrator;

limiting thereby any malicious insider attacks on the network computer system by its own system administrators.

* * * * *